

Описание  
системы



Apache



OpenVZ  
Linux Containers



PostgreSQL

# СЕРВЕР СБОРА И ХРАНЕНИЯ ДАННЫХ АСКУГ

СТМ.425000.0070С

2014

# СОДЕРЖАНИЕ

## 1. Введение

1.1 Наименование и назначение модуля .....	3
1.2 Основные возможности .....	3
1.3 Уровень подготовки пользователя .....	3
1.4 Перечень эксплуатационной документации .....	3

## 2. Назначение и условия применения

2.1 Область применения .....	4
2.2 Программные и аппаратные требования .....	4
2.2.1 Требования к оборудованию сервера .....	4
2.2.2 Требования к программному обеспечению .....	4

## 3. Структура системы

3.1 Описание и принципы работы.....	5
3.2 Подсистема сборки и укладки данных.....	5
3.3 Подсистема хранения данных .....	7

# 1. ВВЕДЕНИЕ

## 1.1 Наименование и назначение модуля

Сервер сбора и хранения данных (ССХД) предназначен для сбора, хранения и обработки технологических данных о работе узлов учета газа. Он представляет собой решение для сбора, хранения и анализа объемов и других показаний различных параметров работы узла учета.

## 1.2 Основные возможности

ССХД обеспечивает выполнение следующих функций:

- прием и укладка информации;
- обработка данных и вычисление итогов;
- хранение информации и ее архивирование.

## 1.3 Уровень подготовки пользователя

Администратор системы должен обладать следующей квалификацией:

- иметь навыки работы в Интернет с помощью браузера Firefox (Chrome);
- знание Linux на уровне продвинутого пользователя;
- уметь пользоваться командной строкой;
- обеспечить надежность и сохранность индивидуальных паролей;
- уметь пользоваться командной строкой;

Диспетчеру достаточно уметь следующее:

- иметь навыки работы в Интернет с помощью браузера Firefox (Chrome);
- обеспечить надежность и сохранность индивидуальных паролей;

## 1.4 Перечень эксплуатационной документации

При эксплуатации системы необходимо руководствоваться также следующими руководствами:

- АРМ администратора сервера сбора и хранения данных (СТМ.425000.008РП)
- АРМ диспетчера сервера сбора и хранения данных (СТМ.425000.009РП)

## 2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

### 2.1 Область применения

Требования настоящего документа применяются при:

- предварительных комплексных испытаниях;
- опытной эксплуатации;
- приемочных испытаниях;
- промышленной эксплуатации.

### 2.2 Программные и аппаратные требования

#### 2.2.1 Требования к оборудованию сервера

Системные требования к оборудованию сервера:

<b>Процессор</b>	с поддержкой технологии виртуализации, с частотой не ниже 2.4 ГГц.
<b>ОЗУ</b>	не менее 12 GB.
<b>Жесткий диск</b>	40 ГБ

#### 2.2.1 Требования к программному обеспечению

Основным требованием к программному обеспечению является **поддержка технологии виртуализации OpenVZ.**

# 3. СТРУКТУРА СИСТЕМЫ

## 3.1 Описание и принципы работы

Сервер состоит из 2 основных подсистем:

- Подсистема сбора и укладки данных.
- Подсистема хранения данных.

Каждая подсистема представляет из себя отдельную виртуальную машину на базе технологии OpenVZ, позволяющий изолировать среду выполнения и тем самым повысить устойчивость работы системы в целом. В качестве операционной системы — Debian Linux.

В свою очередь подсистема сбора и укладки данных включает:

- Веб-сервер Apache, занимающийся сбором данных. Прием по протоколу http в рамках зашифрованных каналов связи. Формат пакета данных для обмена — XML.
- Программа по укладке информации на сервер хранения данных.

В основе подсистемы хранения данных - СУБД PostgreSQL.

**Основной алгоритм работы:**

- Коммуникационное устройство строит зашифрованный туннель (PPT или OpenVPN) с сервером по сбору данных.
- После установления соединения, коммуникатор по протоколу http обращается на сервер и методом POST отсылает пакет данных. Сервер принимает информацию, укладывает ее во временное локальное хранилище и возвращает коммуникатору ответ об успешном приеме.
- Программа по укладке информации в базу данных мониторит временное хранилище и при появлении нового пакета данных, считывает его, переформатирует в нужный для сервера хранения данных вид, и укладывает ее в базу данных. После успешного сохранения на сервере, данный пакет удаляется из временного хранилища.
- На сервере хранения данных производится расчет различных параметров, средних показателей и вычисление итоговых значений за определенный период.

## 3.2 Подсистема сбора и укладки данных

В основе подсистемы — веб-сервер **Apache**. В качестве бэкенда — PHP, Python.

Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами сервера Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д.

**Ядро Apache** включает в себя основные функциональные возможности, такие как обработка конфигурационных файлов, протокол HTTP и система загрузки модулей. Ядро (в отличие от модулей) полностью разрабатывается Apache Software Foundation, без участия сторонних программистов.

**Apache HTTP Server** поддерживает модульность. Существует более 500 модулей, выполняющих различные функции. Часть из них разрабатывается командой Apache Software Foundation, но основное количество — отдельными Open source-разработчиками.

Модули могут быть как включены в состав сервера в момент компиляции, так и загружены динамически, через директивы конфигурационного файла.

В модулях реализуются такие вещи, как:

- Поддержка языков программирования.
- Добавление функций.
- Исправление ошибок или модификация основных функций.
- Усиление безопасности.

Система конфигурации Apache основана на текстовых конфигурационных файлах и имеет три условных уровня конфигурации:

- Конфигурация сервера.
- Конфигурация виртуального хоста.
- Конфигурация уровня директории.

Имеет собственный язык конфигурационных файлов, основанный на блоках директив.

Apache имеет встроенный механизм виртуальных хостов. Он позволяет полноценно обслуживать на одном IP-адресе множество сайтов, отображая для каждого из них собственное содержимое.

Для каждого виртуального хоста можно указать собственные настройки ядра и модулей, ограничить доступ ко всему сайту или отдельным файлам.

Также, существуют модули, позволяющие учитывать и ограничивать ресурсы сервера (CPU, RAM, трафик) для каждого виртуального хоста.

### **Интеграция с другим ПО и языками программирования:**

Существует множество модулей, добавляющих к Apache поддержку различных языков программирования и систем разработки.

К ним относятся:

- PHP (mod\_php).
- Python (mod\_python, mod\_wsgi).
- Ruby (apache-ruby).
- Perl (mod\_perl).

Кроме того, Apache поддерживает механизмы CGI и FastCGI, что позволяет исполнять программы на практически всех языках программирования, в том числе C, C++, Lua, sh, Java.

Apache имеет различные механизмы обеспечения безопасности и разграничения доступа к данным. Основными являются:

Ограничение доступа к определённым директориям или файлам.

- Механизм авторизации пользователей для доступа к директории на основе **HTTP-аутентификации** (`mod_auth_basic`) и **digest-аутентификации** (`mod_auth_digest`).
- Ограничение доступа к определённым директориям или всему серверу, основанное на IP-адресах пользователей.
- Запрет доступа к определённым типам файлов для всех или части пользователей, например запрет доступа к конфигурационным файлам и файлам баз данных.

Существуют модули, реализующие авторизацию через СУБД или PAM.

Также, существует механизм `suexec`, используемый для запуска скриптов и CGI-приложений с правами и идентификационными данными пользователя.

Для реализации шифрования данных, передающихся между клиентом и сервером, используется механизм **SSL**, реализованный через библиотеку **OpenSSL**. Для удостоверения подлинности веб-сервера используются сертификаты X.509.

Существуют внешние средства обеспечения безопасности, например `mod_security`.

### 3.3 Подсистема хранения данных

В основе подсистемы — СУБД PostgreSQL.

**PostgreSQL** - это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных. PostgreSQL поддерживается на всех современных Unix системах (34 платформы), включая наиболее распространенные, такие как Linux, FreeBSD, NetBSD, OpenBSD, а также под Mac OS X. Начиная с версии 8.X PostgreSQL работает в "native" режиме под MS Windows.

Основные возможности:

- **Надежность** PostgreSQL является проверенным и доказанным фактом и обеспечивается следующими возможностями:
- Полное соответствие принципам **ACID** - атомарность, непротиворечивость, изолированность, сохранность данных.
- **Atomicity** - транзакция рассматривается как единая логическая единица, все ее изменения или сохраняются целиком, или полностью откатываются.
- **Consistency** - транзакция переводит базу данных из одного непротиворечивого состояния (на момент старта транзакции) в другое непротиворечивое состояние (на момент завершения транзакции). Непротиворечивым считается состояние базы, когда выполняются все ограничения физической и логической целостности базы данных, при этом допускается нарушение ограничений целостности в течение транзакции, но на момент завершения все ограничения целостности, как физические, так и логические, должны быть соблюдены.

- **Isolation** - изменения данных при конкурентных транзакциях изолированы друг от друга на основе системы версионности
- **Durability** - PostgreSQL заботится о том, что результаты успешных транзакций гарантировано сохраняются на жесткий диск вне зависимости от сбоев аппаратуры.
- **Многоверсионность** (Multiversion Concurrency Control, MVCC) используется для поддержания согласованности данных в конкурентных условиях, в то время как в традиционных базах данных используются блокировки. MVCC означает, что каждая транзакция видит копию данных (версию базы данных) на время начала транзакции, несмотря на то, что состояние базы могло уже измениться. Это защищает транзакцию от несогласованных изменений данных, которые могли быть вызваны (другой) конкурентной транзакцией, и обеспечивает изоляцию транзакций. Основным выигрышем от использования MVCC по сравнению с блокировкой заключается в том, что блокировка, которую ставит MVCC для чтения не конфликтует с блокировкой на запись, и поэтому чтение никогда не блокирует запись и наоборот.
- Наличие **Write Ahead Logging (WAL)** - общепринятый механизм протоколирования всех транзакций, что позволяет восстановить систему после возможных сбоев. Основная идея WAL состоит в том, что все изменения должны записываться в файлы на диск только после того, как эти записи журнала, описывающие эти изменения будут и гарантировано записаны на диск. Это позволяет не сбрасывать страницы данных на диск после фиксации каждой транзакции, так как мы знаем и уверены, что сможем всегда восстановить базу данных используя журнал транзакций.
- **Point in Time Recovery (PITR)** - возможность восстановления базы данных (используя WAL) на любой момент в прошлом, что позволяет осуществлять непрерывное резервное копирование кластера PostgreSQL.
- **Репликация** также повышает надежность PostgreSQL. Существует несколько систем репликации, например, Slony, который является свободным и самым используемым решением, поддерживает master-slaves репликацию. Ожидается, что Slony-II будет поддерживать multi-master режим.
- **Целостность данных** является сердцем PostgreSQL. Помимо MVCC, PostgreSQL поддерживает целостность данных на уровне схемы - это внешние ключи (foreign keys), ограничения (constraints).
- **Модель развития** PostgreSQL, которая абсолютно прозрачна для любого, так как все планы, проблемы и приоритеты открыто обсуждаются. Пользователи и разработчики находятся в постоянном диалоге через мэйлинг листы. Все предложения, патчи проходят тщательное тестирование до принятия их в программное дерево. Большое количество бета-тестеров способствует тестированию версии до релиза и вычищению мелких ошибок.
- **Открытость кодов** PostgreSQL означает их абсолютную доступность для любого, а либеральная BSD лицензия не накладывает никаких ограничений на использование кода.
- **Производительность** PostgreSQL основывается на использовании индексов, интеллектуальном планировщике запросов, тонкой системе блокировок, системой управления буферами памяти и кэширования, превосходной масштабируемости при конкурентной работе.



- **Поддержка индексов:**
  - Стандартные индексы - B-tree, hash, R-tree, GiST (обобщенное поисковое дерево)
  - Частичные индексы (partial indices)
  - Функциональные индексы
- **Планировщик запросов** основывается на стоимости различных планов, учитывая множество факторов. Он предоставляет возможность пользователю отлаживать запросы и настраивать систему.
- **Система блокировок** поддерживает блокировки на нижнем уровне, что позволяет сохранять высокий уровень конкурентности при защите целостности данных. Блокировка поддерживается на уровне таблиц и записей. На нижнем уровне, блокировка для общих ресурсов оптимизирована под конкретную ОС и архитектуру.
- **Управление буферами и кэширование** используют сложные алгоритмы для поддержания эффективности использования выделенных ресурсов памяти.
- **Tablespaces (табличные пространства)** позволяют гибкое использование дискового пространства для хранения объектов системы, что также повышает производительность и масштабируемость.
- **Масштабируемость** основывается на описанных выше возможностях. Низкая требовательность PostgreSQL к ресурсам и гибкая система блокировок обеспечивают его шкалирование, в то время как индексы и управление буферами обеспечивают хорошую управляемость системы даже при высоких нагрузках.
- **Расширяемость** PostgreSQL означает, что пользователь может настраивать систему путем определения новых функций, агрегатов, типов, языков, индексов и операторов. Объектно-ориентированность PostgreSQL позволяет перенести логику приложения на уровень базы данных, что сильно упрощает разработку клиентов, так как вся бизнес-логика находится в базе данных. Функции в PostgreSQL однозначно определяются названием, количеством и типами аргументов.

На рисунке 3.1 приведена ER диаграмма системного каталога PostgreSQL, в котором заложены все сведения об объектах системы, операторах и методах доступа к ним. При инициализации PostgreSQL кластера (команда `initdb`) создаются две базы данных - `template0` и `template1`, которые содержат предопределенный по умолчанию набор функциональностей. Любая другая база данных наследует `template1`, таким образом, часто используемые объекты и методы можно добавить в системный каталог `template1`.

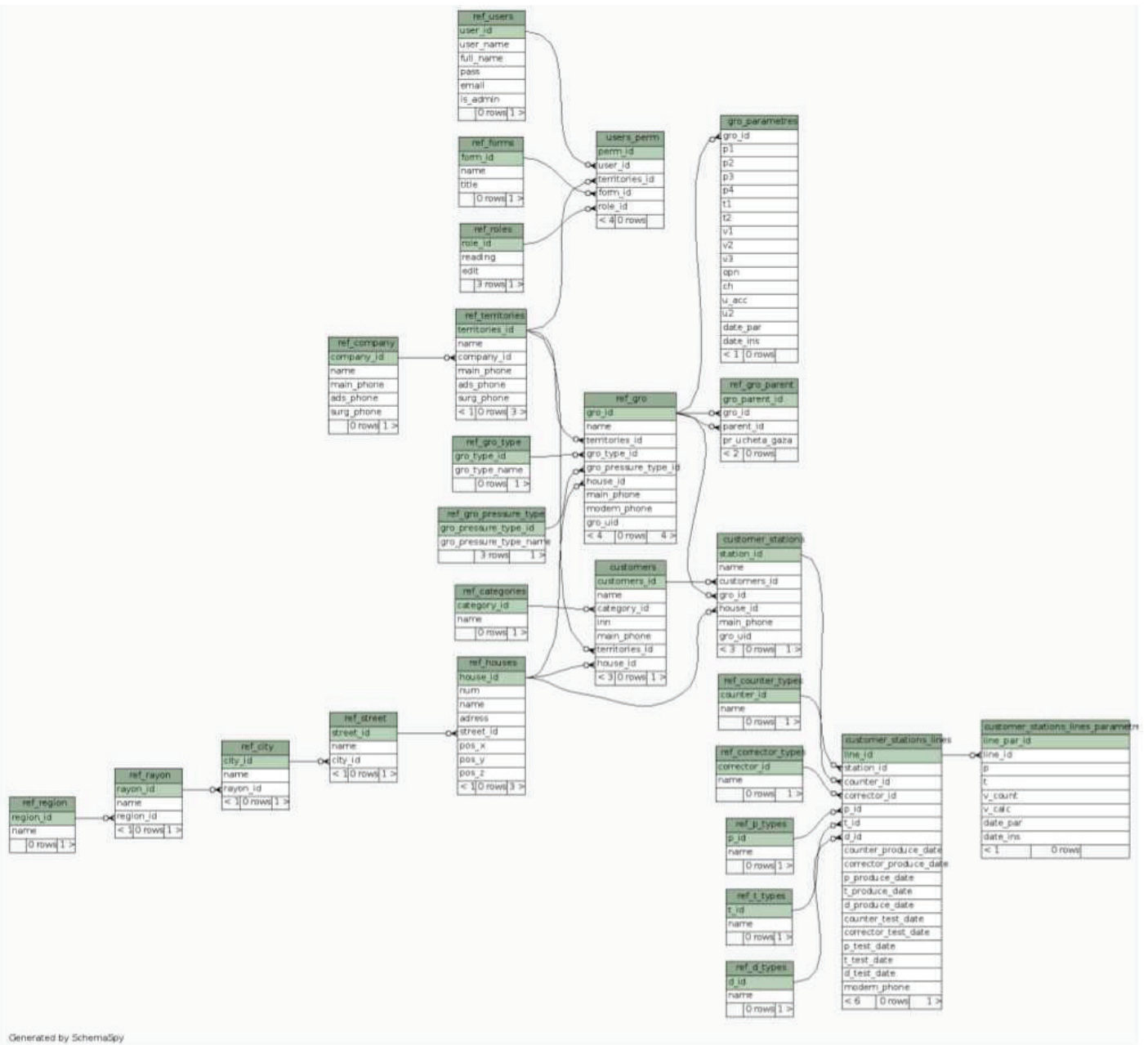


Рисунок 3.1 - Схема БД подсистемы хранения данных.



ООО ЭПО «Сигнал»

4119 Саратовская область, г. Энгельс-19

Тел./факс: 8 (8453) 76-11-11, 75-04-57

Справочная служба (звонок бесплатный): 8-800-100-19-51

Горячая линия по вопросам качества и эксплуатации: 8 (845) 750-425

e-mail: [marketing@eposignal.ru](mailto:marketing@eposignal.ru), [opgo@eposignal.ru](mailto:opgo@eposignal.ru)

[www.eposignal.ru](http://www.eposignal.ru)